We normally publish the post-mortem for an assignment after it has been marked and released. Here is a list of common errors provided by the graders for assignment 4.

**General**

- Many students were missing examples/tests for base case(s) throughout the assignment. When writing a recursive function, there should be an example/test for every base case, and an example/test for at least one of the recursive cases, as discussed in the style guide.

- Types in contracts should always be capitalized. This includes user-defined types as well.

- Where applicable, contract types should be as specific as possible. For example, a function that consumes a Rental should be represented with Rental in the contract, and not (anyof Bicycle Boat Horse).

- Contracts that consumed or produced lists often had formatting errors. For example, if a function consumes a list of numbers, it should be indicated as (listof Num) in the contract **exactly**, and not listof(Num), list of Num, list-num, list, or any other variation.

**Question 2**

- Many students used (anyof Bicycle Boat Horse) instead of Rental in their contract. If there is a user-defined type, contracts should include it everywhere applicable.

- Many students did not define constants for the Rental rates or Rental capacities associated with a Bicycle, Boat, or Horse.

- Many students did not include a contract for their rental-template. As discussed in the style guide, templates should always include a contract.

- Many students had templates that differed greatly from the solutions. The rental-template should look similar to the minfo-template on slide 32 of module 4.

- In part (c), some students did not make sure that a Horse rental was valid only if the consumed number of renters did not exceed the capacity of a Horse, and that the consumed duration did not exceed the Horse’s stamina.

- In part (d), some students did not correctly communicate in their requirements that the Rental must be valid according to the consumed parameters.

- In part (d), some students did not correctly account for the case where the number of renters is greater than 3 for a Horse.
Question 3

- In part (b), some students did not check the case that a list contains only a single integer. e.g. (cons 1 empty)
- In part (c), some students missed the non-empty list requirement for geometric mean.

Question 4

- In part (a), many students wrote (listof Sym) instead of (listof Any) in contract. rainbow? can take any list.
- In part (c), many students failed to add requirement for colour and rainbow. colour is one of 'red, 'orange, 'yellow, 'green, 'blue, 'indigo, 'violet; rainbow is a valid rainbow or define a type Colour
- For all parts, some students had more helper functions than necessary. Check out the posted solution for an alternative. (No mark deducted)

Ongoing Errors

The following is a list of common errors from previous assignments that were still repeated for assignment 4.

- Many students are still missing parameter references in their purpose statements. Purpose statements should meaningfully use each parameter name, and the parameter names should be written exactly as they appear in the function header.
- Constants (and helper functions) should be defined above the design recipe for the function they are used in. Some students are still defining their constants and helper functions between their examples and function definition, or after the main functions specified in the assignment.
- All design recipe components, except for tests, are required for helper functions. Some students did not include these design recipe components for their helper functions.
- For now, equal? should only be used to compare two values of unknown types, or values that can take on more than one type. When the types of the arguments are known, use the most appropriate comparison function (e.g., symbol=?).
- Starting a new cond expression immediately in an else clause is unneeded. Instead, directly check for the next condition in the original cond expression.